CONNEXIONS™

The Interoperability Report

November 1989 Volume 3, No. 11

ConneXions—
The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Advanced Computing Environments, 480 San Antonio Road, Suite 100, Mountain View, California 94040, USA. Phone: 415-941-3399. Fax: 415-949-1779.

ISSN 0894-5926

From the Editor

What a show! As I write this, we've only just started to recover from our largest event to date, INTEROP 89, which was attended by close to 10,000 people. In this issue Daniel Dern gives a brief conference report. A future issue will cover the installation and operation of the Show and Tel-Net in more detail.

As we prepare for our next event, *InfoSec 89*—to be held at the end of this month in Palm Springs, California—Fred Ostapik and April Marine of SRI International report on the *Computer Emergency Response Team* (CERT) workshop which was held in late July. The issues discussed at the CERT workshop will be explored further at InfoSec 89.

With the emergence of low-cost, high-speed dial-up modems, and associated serial line software, IP networks can be constructed using the existing telephone network on an "on-demand" basis. Craig Partridge of BBN Systems and Technologies explains an implementation of dial-up IP service which has been in use by CSNET for some time.

Several important new books were released at INTEROP. Most notably, *The Matrix* by John Quarterman and *The Open Book* by Marshall Rose. Our book review section starts on page 6.

As a followup to an article which appeared in the August 1989 issue of *ConneXions*, Karen Bowers of NRI explains the structure of the IETF and INTERNET-DRAFTS directories.

Dave Chappell presents a tutorial on the OSI Presentation layer in our continuing series *Components of OSI*.

A reminder that all back issues of *ConneXions* are still available. We now also have special *ConneXions* binders for each volume. Binders can be purchased empty ($5.00), or you may wish to take advantage of the special discount price of $100 for any complete volume.

As I was about to finish this month's editorial we were all shook by the strongest earthquake since 1906, measuring 7.1 on the Richter scale. As a result, the production and delivery of this issue has been somewhat delayed. Your patience and understanding is much appreciated. Advanced Computing Environments' headquarters sustained only minor damage, and we resumed normal operations within a couple of days of the disaster.

Response to computer attacks: The CERT Workshop

by Fred Ostapik and April Marine, SRI International

Introduction

The ease with which anyone's computing environment may be violated was made graphic as a result of the infamous "Internet Worm" incident of November 1988. The perpetrator launched software which invaded and disrupted thousands of computers on the Internet. Nor are stand-alone computers immune. They can be invaded by "contaminated" software spread through seemingly innocent exchanges of floppy disks, tapes, and other media.

CERT

The *Computer Emergency Response Team* (CERT) was formed as a direct result of the reactions to the "Internet Worm." Funded by the Defense Advanced Research Projects Agency (DARPA), its main function is to provide a focal point to counteract the effects of computer "intrusions" and to prevent future occurrences. The CERT will accept information relating to network and computer security problems. It forwards the information to experts for disposition, and keeps track of the status until resolution.

The CERT is located in the *Software Engineering Institute* (SEI) at Carnegie Mellon University and has a 24-hour hotline: 412-268-7090. Information can also be forwarded to CERT@SEI.CMU.EDU.

Workshop

At the end of July, 1989, the SEI, CERT, and the *National Institute of Standards and Technology* (NIST), sponsored a workshop on Computer Security Incident Response. The main purpose of the workshop was to establish a framework to help organizations develop their own computer security response centers and coordinate the relationships and communications among these centers to enhance their effectiveness.

Recently, NIST has been cooperating with other agencies, such as the Defense Communications Agency, the Department of Energy, and the National Computer Security Center in an effort to develop a network of response centers similar to the CERT. Each center would serve its own constituency but would also coordinate security activities with the other centers.

Participants

Participants at the workshop were from many varied backgrounds; but they had a common interest—to make computing safer for the users. They all were involved, to some degree, with efforts to identify, contain, and correct computer security problems within a network environment. Some of the participants were already in the process of establishing their own response centers. In addition, vendors, network managers, and legal investigators contributed their unique perspectives to the workshop. This broad representation of diverse constituencies gave eloquent testimony to the complexities involved in counteracting security problems caused by perpetrators exploiting software defects or lax procedures.

Topics

The workshop participants were divided into working groups, each addressing a different facet of response center operations. Among the topics considered were:

- *Incident Handling:* steps required from the identification through the resolution of a computer security problem.
- *Vendor Relations:* effectively involving the vendors in this process.

- *Clearinghouse Activities:* developing and distributing a base of information necessary to resolve current problems and prevent future occurrences.
- *Communications:* establishing solid communications channels for both emergency and routine use.
- *Constituency Relations:* defining the community to be serviced by each response center.
- *Research Issues:* relating to the future enhancements of the response centers.
- *Legal and Investigative Issues:* determining the support provided by law enforcement agencies and the courts.

Goals

The workshop as a whole also focused on several major goals. The most important of these goals was to create a response center charter, which would provide the basic policies and procedures establishing these centers, and give some structure to their relationships with each other, their constituencies, and their vendors.

The necessary condition for an effective charter is to provide sufficient formal structure to prevent administrative chaos, and yet maintain enough flexibility to react swiftly to unanticipated events. It would serve no one's purpose to prevent—swift ad-hoc actions, such as that taken by a number of network "wizards" who so effectively and quickly neutralized the "Internet Worm" last November.

Conclusion

The main operational issues developed in the workshop are:

- What kind of assistance and guidelines should be available to help organizations create their CERT-like facilities?
- What kind of communications structure should be in place to allow these facilities to disperse information to a wide constituency—fax networks, telephone chains, electronic mailing lists?
- What services can federal agencies, such as NSCS, NIST, FBI, make available to these facilities to enhance their effectiveness—publications, training, investigative support?

These and related issues will be discussed in much greater detail during subsequent workshops. The net result of this activity is to produce a national game plan that provides some measure of effective support for the creation and operation of a network of such CERT facilities.

FRED M. OSTAPIK is a Senior Research Analyst at the Network Information Systems Center at SRI International. His tasks include the design and implementation of audit trail systems for the Defense Data Network (DDN). Fred holds an M.S. in Computer Science, and a B.S. in Applied Mathematics and Physics from the University of Wisconsin–Madison.

APRIL MARINE received a B.A. from the University of California, Berkeley, and has been with SRI International, DDN Network Information Center (NIC) since 1985. She has worked on a variety of projects within the NIC, including the establishment of a bibliographic database of protocol-related documents. She has contributed to the NIC's role as a central communications interface between end-users who experience security problems and those agencies striving to solve these problems.

Issues in Dial-up IP Service

by Craig Partridge, BBN Systems & Technologies Corp.

A couple of organizations, most notably CSNET, now offer *dial-up IP service* to their members. Dial-up IP is a novel service in the Internet community, because it implements IP over intermittently connected telephone lines instead of dedicated links. In this article we look at some of the issues involved in implementing dial-up IP, from the perspective of one of the designers of the CSNET dial-up IP software.

Phone line speeds

Indeed, the idea of dial-up IP is fairly old. Members of the CSNET technical staff, most notably Dennis Rockwell, considered the idea of dial-up IP service as early as 1984. The Ballistics Research Laboratory is said to have supported a limited dial-up IP system in the mid-1980s. The problem at that time was that dial-up modems generally had data rates too slow to support acceptable IP performance. What really motivated the development of dial-up IP was the appearance of low-cost 9600 bps modems in late 1987 and early 1988.

Two modes of dialup service

When CSNET was designing its dial-up IP service, it was clear that at least two modes of dial-up IP service were possible:

- *On-Command IP service*, in which a user or program has to issue an explicit command to make a phone call and establish an IP link; and

- *On-Demand IP service*, in which a dial-up link is established if an IP packet needs to be sent, and the link is not already up.

We chose to support both forms of service.

Supporting on-demand IP service was by far the hardest problem, since it gave customers substantially less control over their phone bills. As a result, we had to develop facilities to limit when on-demand phone calls would be made. For example, it is possible to limit the time of day when phone calls are made (to avoid peak-hour calls), as well as the protocol and hosts from which the first IP packet originated (so ICMP *pings* from a student's workstation won't cause a phone to be dialed).

On-command service also had some interesting problems associated with it. The TCP/IP protocol suite is not designed on the assumption that a host's normal state is to be disconnected from the network. The most common, and important, instance of this problem is with e-mail. Suppose you bring up your dial-up IP link for an hour every night at midnight and you want to pick up your mail. How do you make sure that every host on the Internet which has e-mail waiting to send to your system knows that your link is up and that they should send all your mail now?

In theory one might try to use the SMTP TURN command, but the number of Internet hosts (around 150,000 at last estimate) and the fact that TURN is a believed to be a security liability and thus widely unimplemented, make this impractical. Indeed, in practice, there is apparently no good way to notify the Internet when a system of interest has come on-line. So, for e-mail, we implemented a pick-up service. E-mail is routed using MX records to a specific system to be held for pickup (in this case, relay.cs.net).

When a dial-up system connects to the Internet, it uses a special protocol to notify the collection system that it is up. The collection system then starts up its mailer, which delivers all the accumulated mail.

Security

An additional problem was *security*. We didn't want dial-up to be usable by just anybody; we wanted some certainty that when a phone call came in, that the machine at the other end was the machine it claimed to be. After some thought, we decided the simplest way to do this was to force remote systems to log in. Using login for access control also had the advantages that it was a security mechanism system administrators understood, and thus we would not have to spend time justifying a specialized security mechanism to potential customers.

One nuisance of using login as a security mechanism was that we had to find some way for the dial-up network interface (which is below IP in the kernel) to make a phone call and negotiate a login sequence. Clearly we didn't want to put all the dialer and host interface intelligence necessary into the kernel. So we developed a way for the dial-up network interface to ask an application to establish a connection for it. The application contains everything needed to dial the phone and log in. Once the login is complete, the application notifies the dial-up interface that the link is "live," and the interface begins transmitting or receiving IP datagrams.

Miscellaneous Nuts and Bolts

Implementing the dial-up IP interface itself was actually pretty easy. Only three important problems needed to be solved: how to package IP datagrams over the dial-up link and how to deal with failed phone calls and how to deal with idle links.

We chose to use the SLIP protocol (RFC 1055) for encapsulation, because the point-to-point protocol standard was not yet ready.

When a phone link fails we chose to simply hangup the modem and discard any queued IP datagrams. This was simpler than trying to re-establish the link.

If a phone link is idle for a few minutes (no datagrams sent or received) our software again hangs up the modem. This saves money for our users.

Conclusion

Overall performance of the software, which has now been in use for a year, has been quite good. Users generally seem satisfied with the quality of service they receive, and are pleased to be able to FTP documents without having to pay the cost of a full time connection.

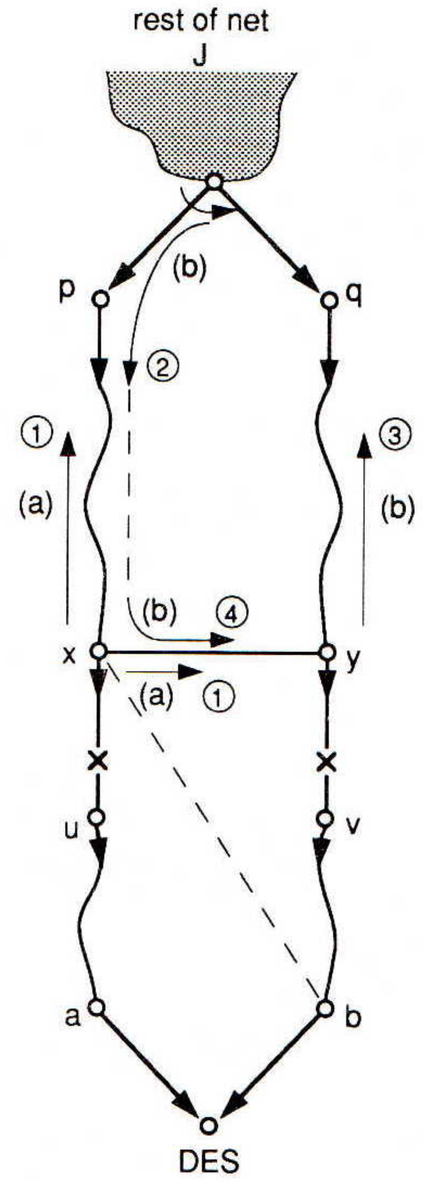
References

"Implementation of Dial-up IP for UNIX Systems," by Leo Lanzillo and Craig Partridge, in Proc. 1989 Winter USENIX Conference.

CRAIG PARTRIDGE received his B.A. (1983) and M.Sc. (1988) from Harvard University and expects to get his Ph.D. from Harvard soon. For the past six years he has worked for BBN Laboratories on a variety of networking related projects including CSNET, the NSF Network Service Center (NNSC), and various projects concerned with distributed systems, IP transport protocols, network management and Gigabit networking. In addition, he is a member of the Internet End-To-End Research Group, the Internet Engineering Task Force, and the Internet Engineering Steering Group. He currently splits his time at BBN between the NNSC and a research project on Gigabit-speed networking. Craig is also the editor of ACM SIGCOMM's *Computer Communication Review*.

Erratum

In the August 1989 issue of *ConneXions* the author had a minor error in the example of Figure 1 on page 10. The link between nodes x and y must be established ***after*** the links (x, u) and (y, v) fail; otherwise, nodes x and y would be juncture nodes themselves, and the labeling scheme of Tsuchiya's algorithm would work. Similarly, the path in dashed lines would have to be established after the link failures. The problem with Tsuchiya's algorithm is that it does not guarantee that the labels stored by a group of nodes that are connected to one another but have no physical path to a destination necessarily correspond to identifiers of nodes in that group.



(1) Bad news propagates to **J** from **x**

(2) **J** uses **q** for alternate path to **DES**

(3) Bad news propagates to **J** from **y**

(4) Erroneous fix propagates to **y** from **J**

Figure 1: Looping with Juncture Nodes

Book Reviews

If you are familiar with the 1984 CCITT recommendations on Message Handling Systems (X.400 or MHS), then *The X.400 Blue Book Companion* is for you. (The book is published in the UK by Technology Appraisals, the ISBN number is 1 871802 00 8.) In a little over 100 pages, it presents a coherent description of the differences between the early 1984 work on MHS and the latest joint ISO/CCITT work on MHS which was finalized in 1988.

1984 versus 1988

There are really only two problems with this book: if you aren't really familiar with the 1984 X.400 work, then while you can glean quite a few things about MHS, then rather than getting the basics about MHS, you'll probably get bogged down in the differences between 1984 and 1988 works. (There is however, a good three page section introducing the Directory, which is one of the nicest I've seen in print.) Second, most chapters contain a section on *Implementation options* which are classic examples of "theory but no practice." That is, these sections contain only the very, very obvious, and aren't any help to the competent implementor.

Overall, if you are implementing MHS, then this book is useful; otherwise, look for something else. —*Marshall Rose*

The Matrix: Computer Networks and Conferencing Systems Worldwide by John Quarterman, Digital Press, ISBN 1-55558-033-5, 719 pp.

Introduction

Throughout modern history great innovations demand great taxonomies. Computer networks are a great innovation of historical magnitude and "The Matrix" now follows as networking's urgent, great taxonomy. The activities surrounding computer networking have gone far beyond computing and are becoming full reflections of human endeavor. To credit them with less would be akin to claiming human language is nothing more than a bag of grammar rules.

The title of the book is taken from William Gibson's cyberpunk novels about the not-so-distant future. The novels describe a vision of a futuristic, world-wide computer network wherein virtually all of human commerce and consciousness plays itself out. Gibson's characters plug themselves into this network via direct biological attachment and enter their life's dramas.

CMC

Quarterman uses the acronym CMC, *Computer Mediated Communications*, to help focus on this network interplay between people. I agree, we need new terminology, CMC it will be.

Although the biological hookups are at least a few months off, The Matrix's introduction assures us the drama is already real:

"Communities of people form around particular networks and topics of discussion supported by networks. Face-to-face conventions have been held and marriages and divorces have been made because of CMC...

...Books (including this one) are researched and reviewed using networks. Scholarly reports composed using computer networks have affected decisions of war and peace and superpower relations." (Introduction, p. 5)

Organization

The book is organized into two major sections. The first section explains the past, present and future of networking's anatomy. The second section provides detailed descriptions of networking's instantiations throughout the world.

The first section gives excellent overviews of a broad range of topics, from computer etiquette and telecommuting to technical explanations of the major networking protocols you are likely to encounter in your on-line travels. Quarterman even shows his exceedingly good taste by quoting me in the History and Futures section, no wonder I like this book!

Atlas

The second section is an exhaustive atlas of the world's computer networks. In here we find everything from the familiar (Internet, EUNET, JUNET) to the obscure (Antarctica is reachable via a Kermit connection over an ATS3 satellite.) From the Halls of Montezuma (try VNET or UNAM) to the shores of Tripoli (Libya, no known networks, well, it is an answer in case you were curious.)

Recommended

You need this book, your workplace needs this book, your library needs this book, your friends, colleagues, students and family need this book. Run, don't walk, to get a copy of The Matrix. —*Barry Shein*

continued on next page

Book Reviews *(continued)*

The Open Book: A Practical Perspective on OSI, by Marshall T. Rose, Prentice-Hall, 1989, 651 pp, including an index and excellent glossary. ISBN 0-13-643016-3.

The following is a glowing review of not just a, but *the*, new OSI book by the Pied Piper of OSI himself. I had the pleasure of working with Marshall as he assembled The Open Book and have, thus, had ample time to go over this conglomeration of OSI knowledge.

Background

OSI is becoming increasingly prevalent in the networking world. Needing to gain knowledge about OSI, the fact, the fiction, and the politics is something that few of us will avoid as this technology matures over the next several years.

Not the first book on OSI but certainly the most comprehensive, The Open Book is a long awaited text addressing OSI and its many aspects which have been largely ignored by other works in the area. The book presents and detailed look at important components of OSI in a style that is packed with practical insight. Its structure is attractive to a wide range of readers. The intent of The Open Book is summarized by these two sentences appearing in its preface:

"This book is about Open Systems Interconnection (OSI). In particular, this book focuses on the pragmatic aspects of OSI: what OSI is, how OSI is implemented, and how OSI is integrated with existing networks."

The content of The Open Book, however, is about much more than just OSI. In order to provide this pragmatic look at OSI the book makes consistent comparisons and analogies of the OSI pieces with the TCP/IP suite of networking protocols. Largely an outgrowth of Marshall's work on the well known ISODE package, the book has the unique characteristics of providing examples, and references to actual software that implements the concepts and is available to the readers.

Audience

Given the breadth of material and the incremental detail, The Open Book is an appropriate text for everyone needing a technical understanding of OSI from the graduate level computer science student, the Presidents and Vice-Presidents of the Silicon Valley's up and coming. It is particularly well suited for those working with TCP/IP networks today and facing the inevitable transition to OSI. Finally, I will go so far as to say that no Engineer working on OSI should be without a copy of The Open Book, it will quickly become the OSI reference of choice throughout the industry.

The book serves as an excellent reference for technical details and implementation guidelines for those dealing with the components of OSI. It is not a business or marketing tool but it does provide an engineering insight into to a part of the TCP/IP and OSI community that players in those aspects of an organization will do well to keep abreast of.

For many, the tangled web of organizations and standards that make up OSI seem to be a hopeless mess. While there may be some truth in that, The Open Book provides an outstanding summary of the what OSI is all about as well as who the players are and the rules they play by.

Content

The bulk of the text is, of course, about the technical aspects of OSI. Most of the people dealing with OSI today will agree that it is the OSI *applications* which are most interesting; they believe this since most already have working networks but see additional value in the functionality of the OSI applications such as X.400 and X.500 to name a few.

Significant portions of the book are dedicated to the OSI Upper Layers which are largely ignored in most other texts on OSI. A bottom up approach to describing the upper layers and finally the applications takes the reader from an understanding of the Session layer (sometimes called the "sewer of OSI") up through many of the OSI applications including Messaging (X.400), Directory (X.500), and a detailed look at the FTAM service. Each component of the upper layer material is concluded with a section describing an actual implementation of this technology.

An overview of the lower layer, or end-to-end, components of OSI is focused primarily on the differences between the various components at each layer and the problems of interconnecting the many choices available with OSI end-to-end services. This section of the text is 100% true to the title in giving a practical perspective on these components rather than a detailed technical look at each piece.

The most valuable portion of the text, for those involved with TCP/IP networks today, will be the discussion of *Transition and Coexistence* strategies. The text presents the motivation for a transition strategy and then follows through with a comprehensive list of options.

Finally, perhaps the least technical part of the text but the most interesting, are the "soap boxes" identifying portions of text which reflect the author's opinion. Within the boxes Marshall takes the opportunity to give the reader insight on the antics of the particular components and the players involved. The soap boxes will have real life value in cutting through some of the mystic surrounding various aspects of the OSI maze.

Organization

The organization and presentation style of The Open Book is clearly one of its many strong components. The clearly defined structure providing an incremental lead in to each portion of the technology is the quality that makes the text so agreeable to a wide range of readers. Those seeking a general overview of OSI, but not the technical nitty gritty, can make their way through the book skipping the very technical sections, which are clearly identified, without missing the flow of the text.

The book provides a excellent introduction to OSI, the players, the pieces, the models, etc; then it jumps into a progressive overview of the OSI end-to-end services, the OSI application services and finally the most comprehensive collection of transition and coexistence from TCP/IP to OSI strategies that you will find in print.

The Final Soap Box

As a finale, the book is concluded with a chapter contained entirely within a "soap box" presenting a perspective on the political ongoings within parts of the TCP/IP and OSI communities. This chapter will not go buy without controversy, it seems to equally offend all but a very few. Reading more like a set of stories from *People* magazine this chapter will prove to be quite an entertaining conclusion for the reader, you might even want to read it first!

—Chris Moore

Highlights from INTEROP 89

by Daniel P. Dern

Ten thousand

Reflecting the growing relevance of interoperability and internetworking, just shy of 10,000 people—9738 to be exact—including over 200 representatives from 50 countries—flocked to ACE's fourth annual INTEROP™ Conference and Exhibition, held this year in the new San Jose Convention Center as well as the function rooms of the nearby Fairmont Hotel. Attendees for the 17 one and two all-day sessions topped 3,000, including a packed-house of 500+ for Dr. Doug "Professor TCP/IP" Comer's double-day intro. The rest barnstormed in for some mix of the three days of shorter conference sessions, Birds of a Feather gatherings, and of course a chance to "stroll the show floor" and see the wares brought by leading and upstart vendors.

Speakers, session leaders and attendees included many founders and current luminaries in the Internet community, from Vint Cerf and Doug Engelbart to Paul Bartoli and IETF Chairman Phill Gross.

Reasons for attending

Conference attendees brought diverse agenda, but nearly all had one focus in common: a user community "back home" with increasing demands for networked computer service. Sample "why I'm here" reasons included:

- "To get a feel for how OSI will take over."
- "To get information about Telnet, FTP, etc."
- "To learn how to manage our new network of Unix systems."
- "To learn about SUN to mainframe connections."
- "To learn about X Windows."
- "We've just installed 250 miles of LAN at our site, and we want to be sure we know how to take care of it."
- "To keep current on issues not discussed in the Internet mailing lists."
- "To get ideas on how to connect the thousands of Macintoshes we're in the process of buying for our site."

SNMP

Session attendees heard advice and updates, like Jim Herman on how internetworking is replacing system integration for organizational strategies—and that "OSI is the best thing that ever happened to TCP/IP." Drs. Jeffrey Case and Vint Cerf press conferenced the formal roll-out of SNMP (Simple Network Management Protocol) based systems, lauding the "tremendous cooperation among the R&D community, the users and the vendors."

Announcements

Bob Braden and Craig Partridge held up the final draft of the TCP/IP Host Requirements RFCs. [See page 23] And IETF Chairperson Phill Gross formally announced the OSPF (Open Shortest Path First) and PPP (Point to Point Serial Line Protocol) protocol specs, seen as the foundation for multivendor gateway internetworking. And while laser lights played through a CO_2 fog, ACE founder Dan Lynch danced up a storm at the Tuesday night opening party.

The exhibits

The INTEROP 89 Exhibition floor provided an equally stirring perspective on the rapid growth of interoperability and internetworking.

In keeping with last year's policy, all relevant exhibiting vendors were required to connect up to the show floor network, demonstrating connectivity and interoperability with all the other connected equipment.

Planning and preparation for this began months before the show. Many vendors pre-tested their equipment across the Internet, and an optional (but highly recommended) "Connectathon" was held the week before the show, to help exhibitors—particularly those participating in special demos—get the bugs out before the show opened.

The advance prep was so successful, in fact, that the Connectathon needed under a day, instead of the up to three days scheduled. Several days before exhibitor equipment arrived, ACE and other staff began laying the wiring for the Show and Tel-Net. Seven physical media were used, from thick and thin Ethernet to microwave and FDDI. Over five miles of cable was used for this year's network.

"The fact that such a complex network can be built in so short a time, connecting so many different vendors, shows that interoperability is no longer a dream for the future; it is a reality today." said Doug Humphrey of Tandem Computers.

IMP 1

Symbolically, IMP 1—the first packet switching node originally installed on the Arpanet, at UCLA twenty years earlier, was the first to arrive on the floor. (And, perhaps with equal symbolism, the machine was soon "put out to pasture" on display outside, presaging the coming retirement of the original ARPANET in favor of the proposed National Research and Education Network superproject.)

Demos

By Wednesday morning, the show floor included:

- Scores of TCP/IP-based systems (over 500 hosts in all)
- 25 vendors showing SNMP network management products
- LAN nodes in over 24 different booths monitored via SNMP
- OSI products from thirteen vendors, including a multi-vendor OSI sub-net
- Eleven companies, staging the largest FDDI event to date

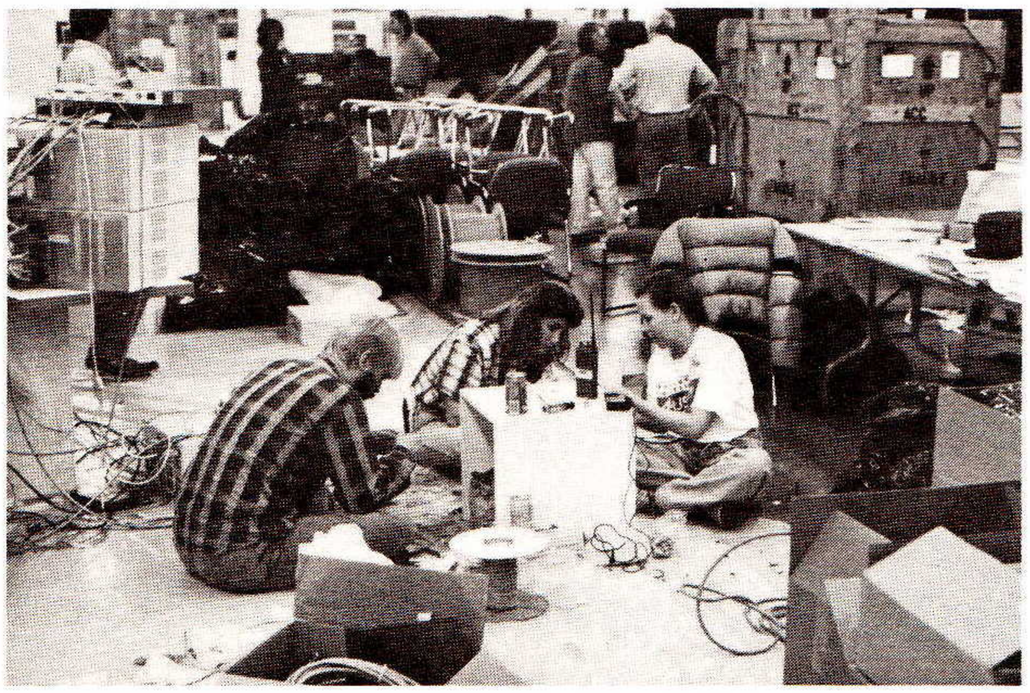
Plus CMOT, X Windows, dual-stack TCP/IP-OSI implementations, TCP/IP running on PCs, implementations of X.400 e-mail, and demonstrations of X.500 Directory Services. Also featured was a real-time demo of the new Point to Point Protocol (PPP).

E-mail centers

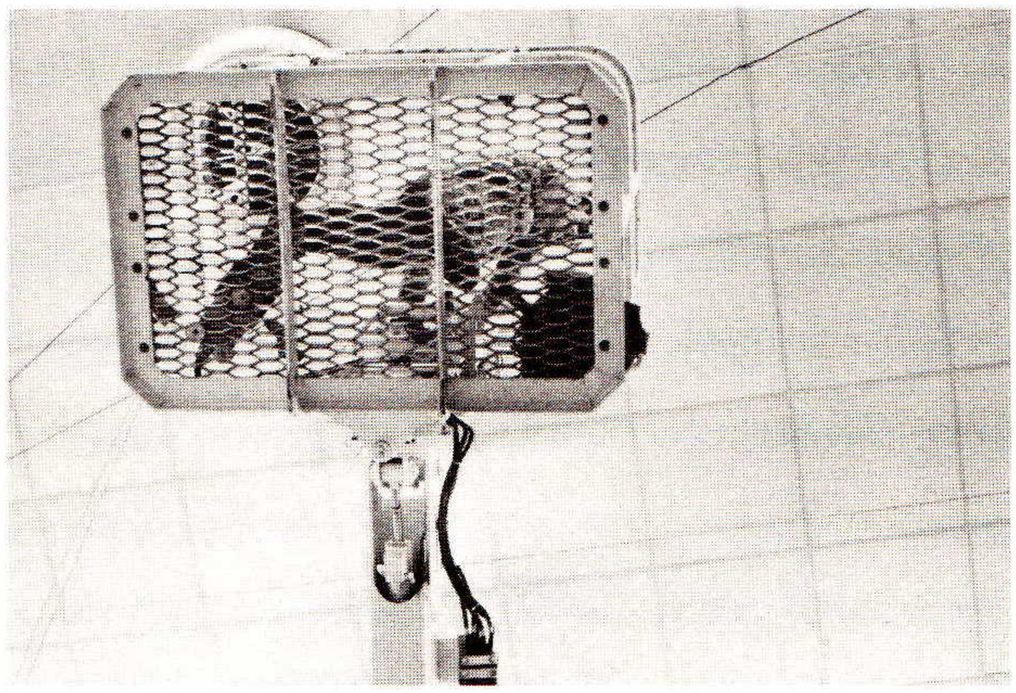
Three clusters of terminals in the Convention Center plus another linked by microwave at the Fairmont Hotel let attendees Telnet to their distant home hosts across the country, to read mail and stay in touch. Show and Tel-Net participants' claim to interoperability and connectivity was further put to the test as many of us walked into booths at random to perform similar activities.

What this year's INTEROP makes clear is that internetworking and interoperability is no longer just the academic interest of the research community. Like PCs, like LANs, these issues have made it into the vendor and business user sectors. By the time INTEROP 90 rolls around, there'll be a whole new set of landmarks and changes—and yet more just-hatching developments to watch as well, no doubt!

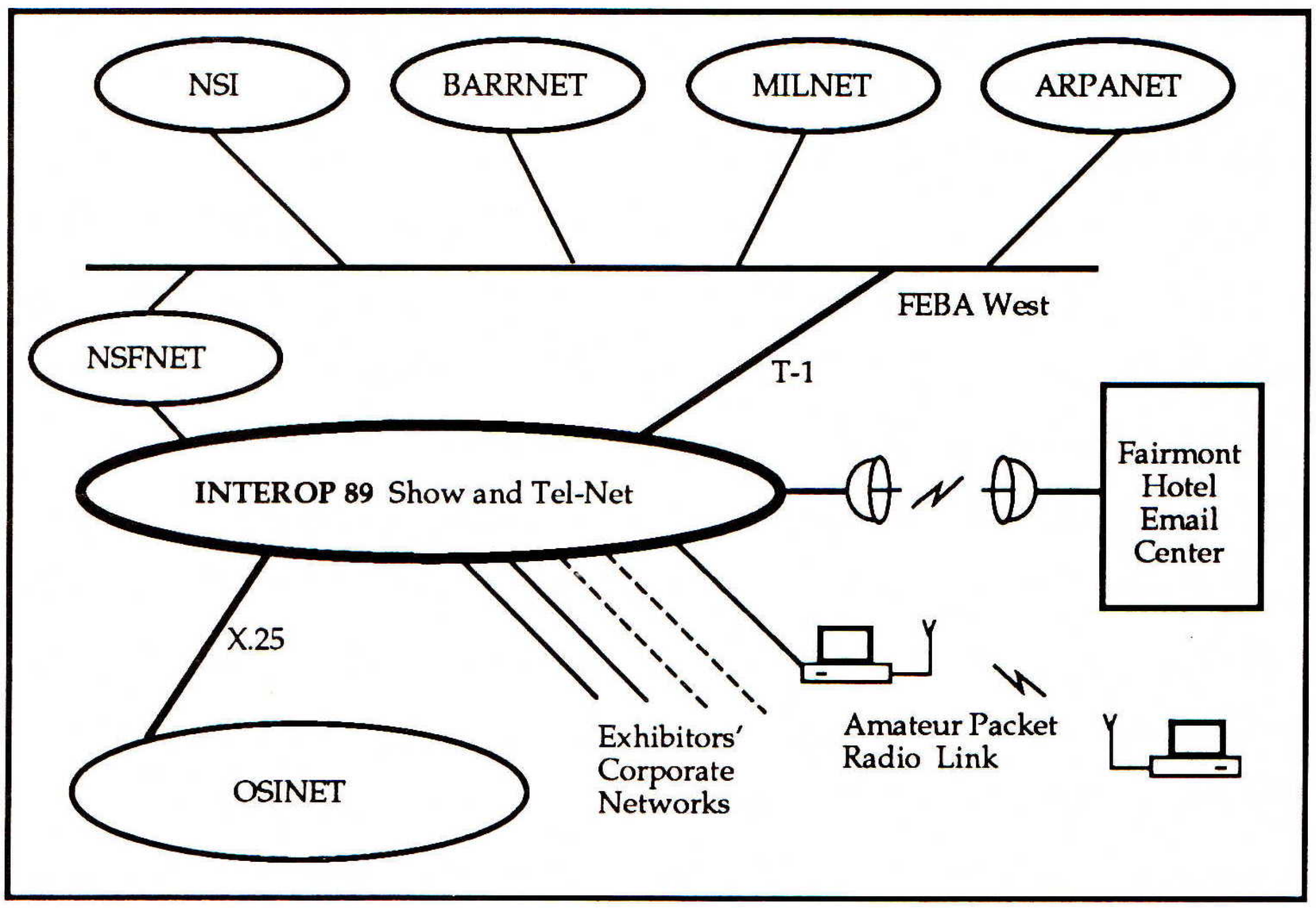
[See the following pages for some images from INTEROP 89].——>



The installation phase—five days of hard work



One of the e-mail centers



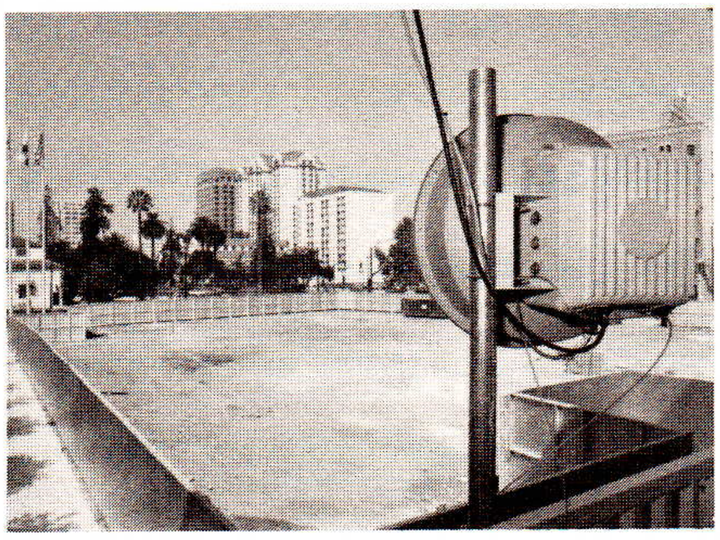
Plenty of connections to the outside world...

INTEROP 89™

```
%last ole

ole    ttyp0    Router4.ShowNet.  Fri Oct  6 15:44 - 15:45   (00:00)
ole    ttyp8    MAIL-MAC-1.ShowN  Fri Oct  6 08:34 - 08:37   (00:03)
ole    ttypc    Router4.ShowNet.  Fri Oct  6 08:25 - 08:26   (00:01)
ole    ttyq7    annex-fairmont.S  Thu Oct  5 20:07 - 20:11   (00:03)
ole    ttyr9    K3MC.ShowNet.COM  Thu Oct  5 10:50 - 10:53   (00:02)
ole    ttyp0    Router4.ShowNet.  Thu Oct  5 08:36 - 08:49   (00:12)
ole    ttyp4    130.128.71.10     Wed Oct  4 20:25 - 20:25   (00:00)
ole    ttyp3    annex-west.ShowN  Wed Oct  4 07:50 - 07:54   (00:03)
ole    ttypb    130.128.254.14    Tue Oct  3 22:31 - 22:44   (00:13)
ole    ttyp9    130.128.254.13    Tue Oct  3 14:59 - 15:01   (00:02)
ole    ttyqc    130.128.254.12    Tue Oct  3 13:40 - 13:42   (00:01)
```

Proof that I was there?



Above: *A Microwave link from the San Jose Convention Center to the Fairmont Hotel was used to operate an off-site e-mail center.*

Right: *The ACE Network Operations Center, hub of the show network.*

Below: *Strollin' the floor.*

Photographs by Daniel Dern, Peter Diggs and Ole Jacobsen.

On-Line Directories: IETF and Internet-Drafts

by Karen Bowers,
Corporation for National Research Initiatives

In the August issue of *ConneXions*, the Internet Engineering Task Force (IETF) announced the availability of two newly revised directories installed at NIC.DDN.MIL: "IETF:" and "INTERNET-DRAFTS:." Since then, shadow directories have been placed at NNSC.NSF.NET for the convenience of the east coast Internet community. (The directory names are not quite the same: "IETF" and "Internet-Drafts." The colon is placed in the NIC.DDN.MIL-installed directories' names to accommodate the TOPS 20 system employed.)

Purpose

The "IETF:" directory has been established as an aid to both veteran IETF members and newcomers. It is comprised of files containing: a general description of the IETF (history, organization, goals); a description of the Internet Activities Board; a summary of active Working Groups within the IETF; IETF meeting dates/locations; upcoming meeting information and an associated RSVP form; the upcoming meeting agenda; and a README file with an overview of directory contents. In addition, individual files have been dedicated to each Working Group and their particular activities. These files contain respective Charters, Status Updates and Current Meeting Reports. The WG files are named in the following fashion:

```
<WGNAME>.charter
<WGNAME>.status
<WGNAME>.report
```

Naming Scheme

The "INTERNET-DRAFTS:" directory presents drafts for review and comment. It contains documents that will be submitted to the RFC Editor for consideration, or will be simply discarded when their purpose has been served. Comments are welcomed and should be addressed to the responsible persons whose names and email addresses are listed on the first page of the respective draft. Each Internet-Draft is placed in a separate file; the following standard naming scheme is used:

File Format	Naming Scheme
ASCII text	DRAFT-<TFNAME>-<WGNAME>-<ABBREVTITLE>-<REVNO>.TXT
PostScript	DRAFT-<TFNAME>-<WGNAME>-<ABBREVTITLE>-<REVNO>.PS
UNIX compressed ASCII	DRAFT-<TFNAME>-<WGNAME>-<ABBREVTITLE>-<REVNO>.TXTZ
UNIX compressed PostScript	DRAFT-<TFNAME>-<WGNAME>-<ABBREVTITLE>-<REVNO>.PSZ

If the document is not being authored in a Task Force, then the author's name will be substituted for the Working Group name (WGNAME) and an organizational affiliation will be submitted for the Task Force name (TFNAME). Example:

```
DRAFT-<ORG>-<AUTHORNAME>-<ABBREVTITLE>-<REVNO>.TXT
```

Recent Drafts

Drafts recently installed include the following:

<DRAFT-IETF-ALERTMAN-ASYNCALERTMAN-00.TXT>	Managing Asynchronously Generated Alerts
<DRAFT-IETF-AUTH-IPOPTION-00.TXT>	The Authentication of Internet Datagrams
<DRAFT-IETF-OSPFIGP-SPEC-07.PS>	Draft OSPF Specification
<DRAFT-IETF-PDN-CLUSTERSCHEME-00.TXT>	Internet Cluster Addressing Scheme
<DRAFT-IETF-PDN-PDNCLUSTER-00.TXT>	Application of the Cluster Addressing Scheme to X.25 Public Data Networks and Worldwide Internet Network Reachability Information Exchange
<DRAFT-IETF-PDN-PDNCLUSTERNETASSIGNM-00.TXT>	Assignment / Reservation of Internet Network Numbers for the PDN-Cluster
<DRAFT-IETF-PERFCC-GWCC-00.TXT>	Gateway Congestion Control Policy

The Internet-Draft directory also contains a README file and an Index-Abstract file to aid the reader in locating drafts of interest.

Getting IETF files

As stated earlier, both the IETF and Internet-Drafts directories are available on-line at NIC.DDN.MIL (west coast) and NNSC.NSF.NET (east coast) and can be accessed by anonymous ftp. The "ls" or "dir" command will permit a review of what files are available and the specific naming scheme to use for a successful anonymous FTP action. For more information, contact: ietf-request@venera.isi.edu.

Reminder: *InfoSec™ 89*

InfoSec™ 89—Practical Perspectives on Computer and Network Security will be held November 28-30, 1989 at the Desert Princess Doubletree Hotel in Palm Springs, California. The seminar is sponsored by Advanced Computing Environments and SRI International.

Format

InfoSec is a unique, three day seminar dedicated to improving awareness of network and computer security issues among MIS Managers and Networking Professionals who are not experts in the area of security. InfoSec provides a one-day tutorial and two days of technical seminars presented by leading experts from the commercial, government, vendor and research communities.

For more information on InfoSec, or to register, call Advanced Computing Environments at 415-941-3399 or FAX at 415-949-1779.

Components of OSI: The Presentation Layer

by David Chappell

Introduction

The presentation layer, layer six in the OSI architecture, plays an important role in solving the problem of open networking. Although actually quite simple, this layer's function is often misunderstood. This is due in part to the layer's somewhat misleading name: "presentation" seems to imply that this layer must be responsible for determining how data is "presented" to a human user, e.g., on a terminal screen. In fact, this is not the case (how data is actually shown on a terminal screen is left to each local system in the OSI architecture). A better name for this layer might have been the "representation" layer, since this more accurately describes its function. This function can be stated very simply: the presentation layer is responsible for determining how all data exchanged by its users (i.e., by application entities) will be represented while in transit across the network.

Concepts

The need for the presentation layer stems from the heterogeneous computing environment for which OSI is intended. Because different computer systems represent information in different ways, some common representation must be agreed upon before that information can be exchanged. For example, IBM 370 series computers represent characters using EBCDIC, while most other computers use ASCII. To transfer a file of characters from an IBM 370 to an ASCII system, a common representation must be used during the actual transfer. This representation may be EBCDIC, ASCII, or something else. Similarly, integers, floating point values, and other kinds of information may be stored internally in a variety of ways. Some common format must be agreed to before this information can be exchanged. It is the job of the presentation layer to provide a mechanism for reaching this agreement.

Translation

If information is represented differently on two communicating systems, one or both must translate from the local form into and out of the standard representation agreed to for communication. In an actual implementation, this may be done by the same software which implements the presentation layer protocol or by software implementing an application layer protocol or both. Where this translation takes place within a computer system is not visible to the outside world (as long as it *does* occur somewhere). It is therefore a local issue and is not subject to standardization.

The presentation layer service and protocol are specified in ISO 8822 and ISO 8823, respectively. Technically identical text is also published as the 1988 CCITT Recommendations X.216 and X.226.

Translating characters from ASCII into EBCDIC may seem like a relatively simple problem. The presentation layer, while providing a means to solve this problem, also allows for solving the more general problem of mapping between differing representations of a much broader range of information. Toward this end, the presentation standards define several abstractions to aid in the solution.

Types and values

In thinking about the presentation layer's problem, it is useful to ask: what aspects of the transferred information are preserved? Clearly it is not the actual bits, as they may be changed, e.g., from ASCII to EBCDIC. What is always preserved, however, is the *type* and *value* of the information transferred.

In the context of OSI's presentation layer, a type may be defined (very informally) as a collection of values distinguished for some reason. Common examples of types includes characters, integers, and floating point (or real) numbers. Every possible value may be considered to be of some type. The value 1, for instance, may be of type "integer," while the value 'A' is of type "character." When information is transferred from one system to another, its representation may change, but its type and its value must be preserved.

Grouping types and values

The example types mentioned so far are all relatively simple. If we admit more complex types into our purview, the problem becomes more difficult. Record types, for example, may be constructed by combining several other types into a single structure. Types of this sort are supported by most high level programming languages. By combining existing types in various ways, a potentially infinite group of types may be created.

Abstract syntaxes

Before two application entities may exchange any information, their supporting presentation entities must reach agreement on how values of all possible types of this exchanged information will be represented. Ideally, every presentation entity would be able to understand and represent all values of every possible type. Because the set of possible types is infinite, however, this is not possible. Instead, types which will be used for a particular instance of communication are grouped into one or more *abstract syntaxes*.

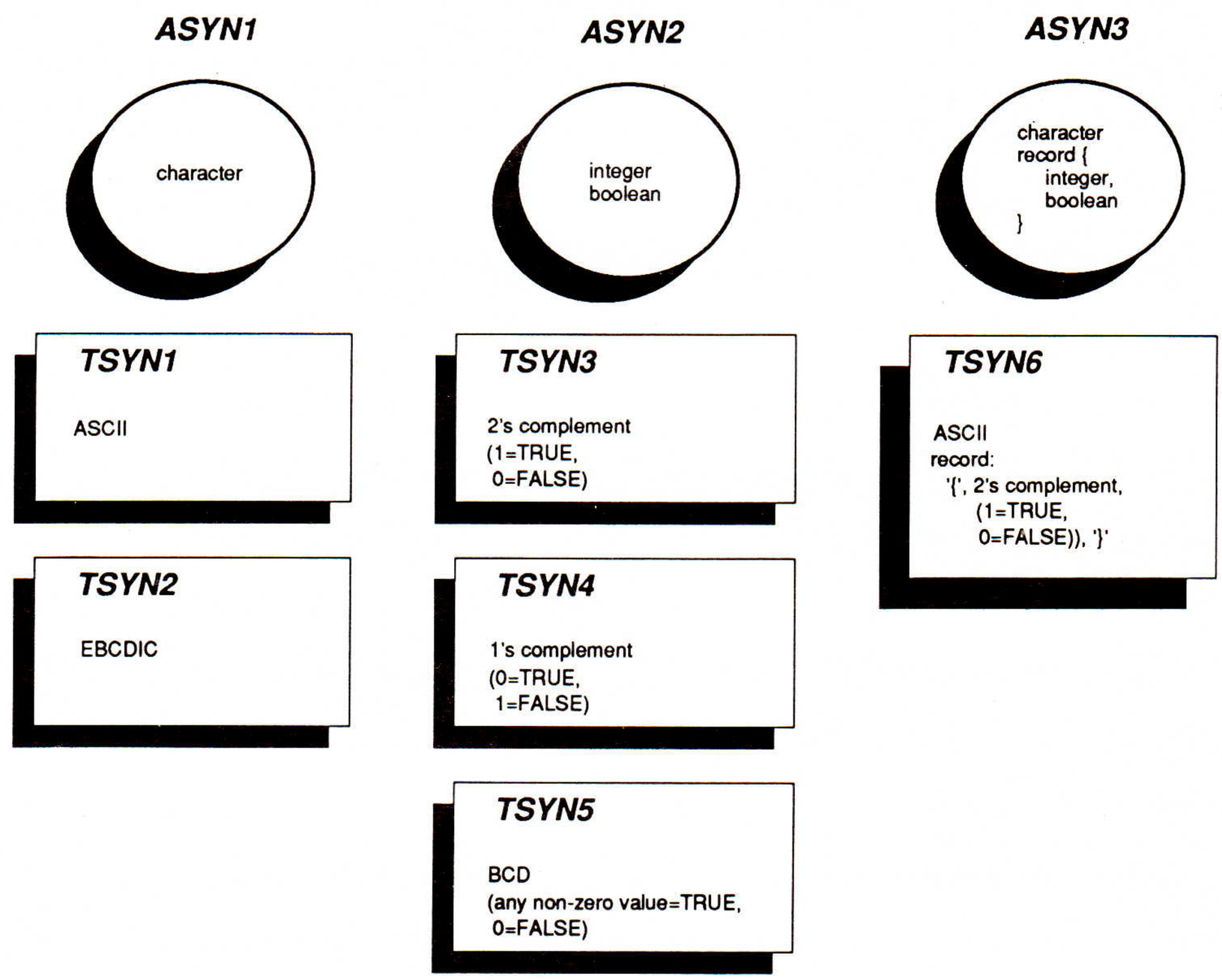


Figure 1: Abstract and Transfer syntaxes

An abstract syntax can be informally thought of as a named group of types. The actual definition, given in ISO 8822, is "the specification of application layer data or application-protocol-control-information by using notation rules which are independent of the encoding technique used to represent them."

continued on next page

The Presentation Layer *(continued)*

As this more formal definition suggests, an abstract syntax merely defines its constituent types—it does not specify how to represent values of those types.

Examples of abstract syntaxes

An abstract syntax may be very simple or very complex. For instance, the abstract syntax called ASYN1, shown at the upper left of Figure 1, contains only the single type "character." The next example, ASYN2, is more complex, containing the types "integer" and "boolean," while the third example, ASYN3, is still more complex, including "character" and a record type containing "integer" and "boolean" elements. (The names assigned to abstract syntaxes are not really character strings; for the moment, however, this simplification is made.)

Note that the same type may occur in more than one abstract syntax—"character," for example, appears in both ASYN1 and ASYN3. Note also that how types are grouped into abstract syntaxes is somewhat arbitrary. We could have combined, say, ASYN1 and ASYN2 into a single abstract syntax, one containing the types "character," "integer," and "boolean." How types are grouped into abstract syntaxes depends on the application at hand. One final point: each type shown in Figure 1 is specified quite informally. In most OSI abstract syntaxes defined so far, the types are specified in a formal language called *Abstract Syntax Notation One* (ASN.1). ASN.1 will be further described in a future issue of *ConneXions*.

Transfer syntaxes

While describing the types to be used by the application entities is useful, it is not enough. As indicated earlier, the primary responsibility of the presentation layer is to determine how values of those types are to be represented during communication. This is achieved by agreeing on a *transfer syntax* for each abstract syntax. A transfer syntax can be thought of as a set of rules for encoding values of some specified group of types, i.e., of some abstract syntax. The term is also sometimes used to describe the actual bit-level representation which results from applying those rules to a particular value.

It is generally possible to define several different transfer syntaxes capable of encoding values of the types contained in any abstract syntax. In Figure 1, two possible transfer syntaxes are shown for the abstract syntax ASYN1. The first, here called TSYN1 (although, as with abstract syntaxes, transfer syntax names are not really character strings), specifies that values of the single type in ASYN1 should be encoded using ASCII. The second possibility, TSYN2, specifies that ASYN1's character values should be encoded using EBCDIC. For a transfer syntax to be usable with an abstract syntax, it must be capable of encoding the values of all types contained in the abstract syntax.

Similarly, values of the types contained in ASYN2 could be encoded using either TSYN3, TSYN4, or TSYN5. For ASYN3, only a single transfer syntax, called TSYN6, is given. Note that, once again, how the encoding rules are grouped together into transfer syntaxes is somewhat arbitrary. While TSYN1 and TSYN2 obviously could not be combined, since they contain two choices for encoding values of a single type, it would be perfectly reasonable to combine TSYN1, TSYN3, and TSYN6 into a single transfer syntax called, say, TSYN99. This new transfer syntax could then be used with any or all of the abstract syntaxes ASYN1, ASYN2, or ASYN3.

Presentation contexts

For each abstract syntax which an application entity wishes to use, exactly one transfer syntax must be selected. (This selection occurs during establishment of the presentation connection, as will be seen later in this article). Each negotiated abstract syntax/transfer syntax pair is called a *presentation context*. All data transferred between two users of the presentation layer (i.e., between two application entities) is contained within some presentation context. The set of presentation contexts which is available at any given time on a presentation connection is known as the *defined context set* (DCS).

Typically, the DCS will contain at least two presentation contexts. For some application layer protocols, or for combinations of those protocols, the DCS may become much larger. To see why this is so, we need to look beyond the simplistic examples of abstract syntaxes given earlier to more realistic examples. One very common example of an abstract syntax is the set of PDUs defined by an application layer protocol. Recall that *all* data exchanged between two presentation users must be part of some presentation context, and thus must be described with an abstract syntax. Perhaps the most important data exchanged by application entities are the PDUs which comprise a particular application protocol, such as FTAM. Each PDU in a particular protocol can be thought of as a value of some type, analogous to a record or structure. The set of all PDUs in some particular application layer protocol, then, comprises a group of types, and thus an abstract syntax. A transfer syntax must be selected for this abstract syntax, and so a presentation context is created.

A second example of an abstract syntax is one describing the data transferred by application entities. Again using FTAM as an example: the contents of every file accessed or transferred by FTAM must be described with an abstract syntax. If the file contains just characters, for example, the abstract syntax ASYN1 from Figure 1 might be used. Since each abstract syntax requires a transfer syntax, each accessed file could potentially have a different presentation context. In any case, distinct presentation contexts are required for the FTAM PDUs and the file's contents.

In some cases, it may not be possible to accurately determine which presentation context is in effect. A *default context* may be defined for these situations. This default may either be negotiated during connection establishment or may by agreed upon *a priori*.

Context management

The initial contents of the DCS are agreed upon during establishment of the presentation connection. In some cases, however, the presentation users may not know all the abstract syntaxes they will need at the time the connection is established. Two users of the FTAM protocol, for example, must agree on an abstract syntax for each file to be transferred or accessed. But they might not learn a file's abstract syntax until the file is opened, which occurs *after* the presentation connection is established. To allow for this possibility, the presentation layer provides *context management*.

The service provided is just what it sounds like: the presentation users can manage, i.e., add to and delete from, the current DCS. Either user can give the presentation layer a new abstract syntax and request it to negotiate a transfer syntax capable of representing that abstract syntax's types.

continued on next page

The Presentation Layer *(continued)*

In other words, either user can request the creation of a new presentation context. Similarly, either user can request the deletion of an existing presentation context from the DCS.

The presentation protocol

The service provided by the presentation layer to its users is quite simple: they are allowed to establish a presentation connection, transfer data across it, and release that connection. Optionally, they may modify the defined context set through context management.

One other group of services is also provided by the presentation layer. These provide a direct pass-through of the session layer's services, allowing an application entity to access those services. This aesthetically unappealing but necessary function occurs because, although the presentation entity is the direct user of all session services, it is not intelligent enough to know how and when to use them. Instead, the application entity (or sometimes the application process itself) must decide when the protocol tools provided by the session layer are used. Because the OSI architecture doesn't allow skipping layers, the presentation service definition contains these pass-through services, allowing application layer access to session services without (technically) violating OSI's strict layering rules.

Establishing connections

Before any information exchange may take place, both systems must agree on which abstract and transfer syntaxes are to be used. This agreement is reached during the establishment of the presentation connection.

Connection initiator

When one application entity wishes to communicate with another, it must first establish a presentation connection between the two systems. To do this, it issues a P-CONNECT request primitive to its supporting presentation entity (see Figure 2). Among the various parameters associated with this primitive is the list of abstract syntax names which it wishes to use on this connection. Only the names of predefined abstract syntaxes may be passed; it is not possible to define a new abstract syntax dynamically.

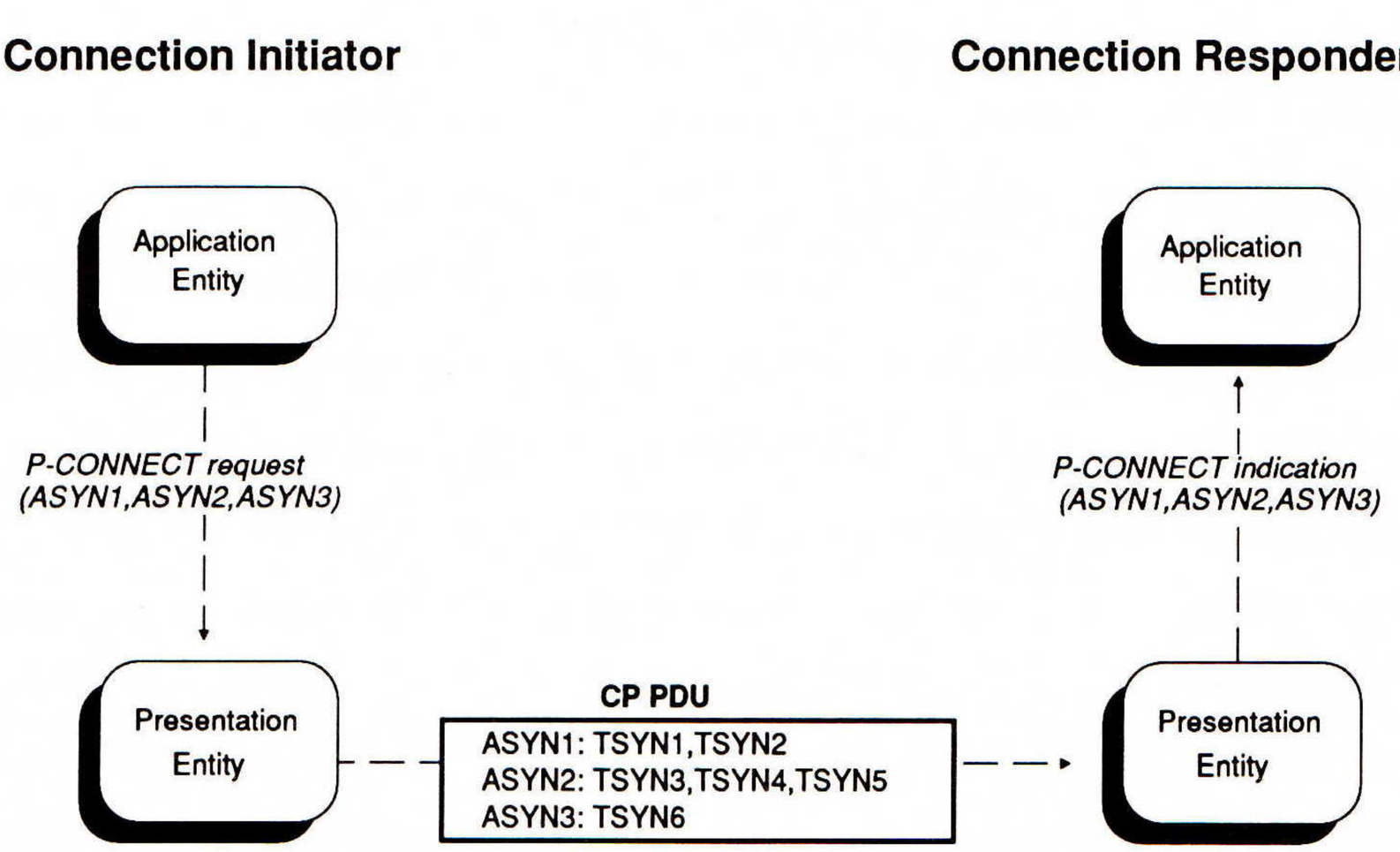


Figure 2: Establishing a presentation connection

An application entity could, for example, issue a P-CONNECT request indicating the abstract syntaxes ASYN1, ASYN2, and ASYN3.

The supporting presentation entity which receives this P-CONNECT request must determine which transfer syntaxes it can use to represent information from each of the specified abstract syntaxes. It then sends a CP PDU to its peer presentation entity indicating both the requested abstract syntaxes and the transfer syntaxes it is willing to support for each abstract syntax. (Any necessary lower layer connections must also be established before this PDU is sent.) For instance, transfer syntaxes TSYN1 and TSYN2 may be listed for ASYN1, TSYN3, TSYN4, and TSYN5 for ASYN2, and TSYN6 for ASYN3.

(An important note: this is an architectural description. Although the protocol exchanges must remain unchanged, an implementor is free to implement the presentation/application layer interface any way he or she chooses. In fact, a real implementation isn't even required to contain a neat software interface between the two layers.)

Connection responder

Upon receipt of a CP PDU, the responding presentation entity must issue a P-CONNECT indication primitive to the application entity above (also shown in Figure 2). It includes among the parameters of this primitive the names of the abstract syntaxes contained in the CP PDU. The responding application entity then responds with a P-CONNECT response primitive (see Figure 3), again containing the names of the abstract syntaxes it is willing to use. These abstract syntax names must include all or a subset of those found on the P-CONNECT indication. Note once again that only *names* are used for both abstract and transfer syntaxes; no mechanism currently exists for describing new syntaxes during connection establishment.

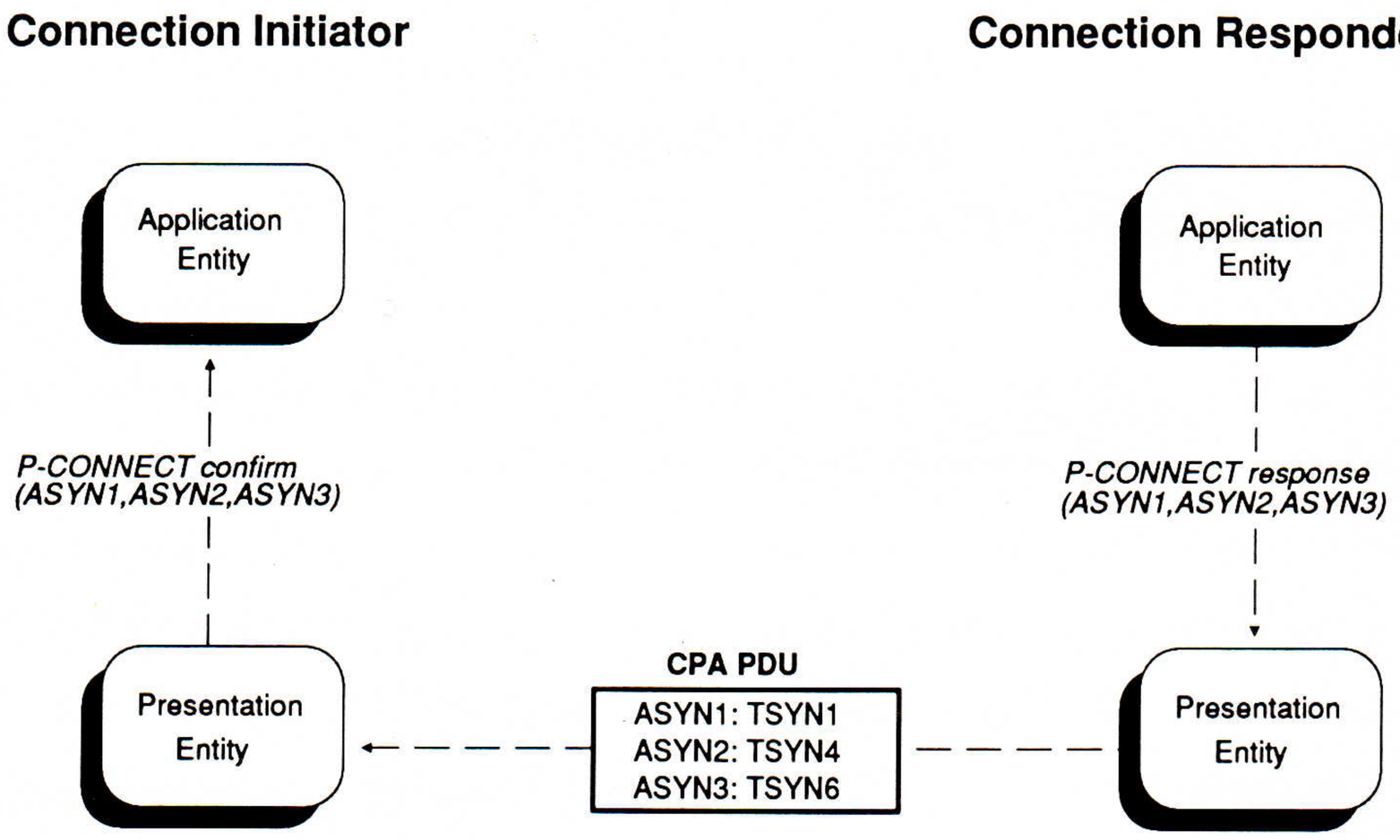


Figure 3: Establishing a presentation connection (continued)

continued on next page

The Presentation Layer *(continued)*

The responding presentation entity, after receiving the P-CONNECT response accepting the presentation connection, must build and send a CPA PDU. On this PDU it indicates which of the proposed transfer syntaxes it has chosen for each selected abstract syntax. For example, the responding presentation entity may choose TSYN1 for ASYN1, TSYN4 for ASYN2, and TSYN6 for ASYN3.

The initiating presentation entity must accept these choices. It also, upon receipt of the CPA PDU, indicates the chosen abstract syntaxes as parameters on the final primitive in this sequence, P-CONNECT confirm, which is given to the initiating application entity. Once this exchange is complete, several things have been accomplished:

- A presentation connection has been established;
- The application entities have negotiated which abstract syntaxes they wish to use, i.e., they have agreed on the types of data they will exchange during this communication;
- The presentation entities have negotiated a representation, a transfer syntax, for each abstract syntax selected by the application entities.

Transferring data

All data transferred in presentation PDUs is always encoded using the agreed transfer syntax for the data's presentation context. In general, each unit of information to be transferred is considered a presentation data value (PDV). The type of this value must appear in at least one of the abstract syntaxes available in the defined context set. PDVs are then grouped into PDV-lists, each containing one or more PDVs from the same presentation context (and thus the same abstract syntax), encoded according to that context's transfer syntax. Each encoded PDV-list begins with a presentation context identifier (PCI), a unique integer value assigned to each presentation context during connection establishment. The PCI allows the receiver to identify this data's presentation context and thus correctly decode the incoming information. A single instance of presentation user data may contain several encoded PDV-lists, and thereby convey information in several different presentation contexts at once.

A simplified diagram of this appears in Figure 4. Two encoded PDV-lists are shown, each with its own presentation context identifier (PCI) and encoded presentation data values.

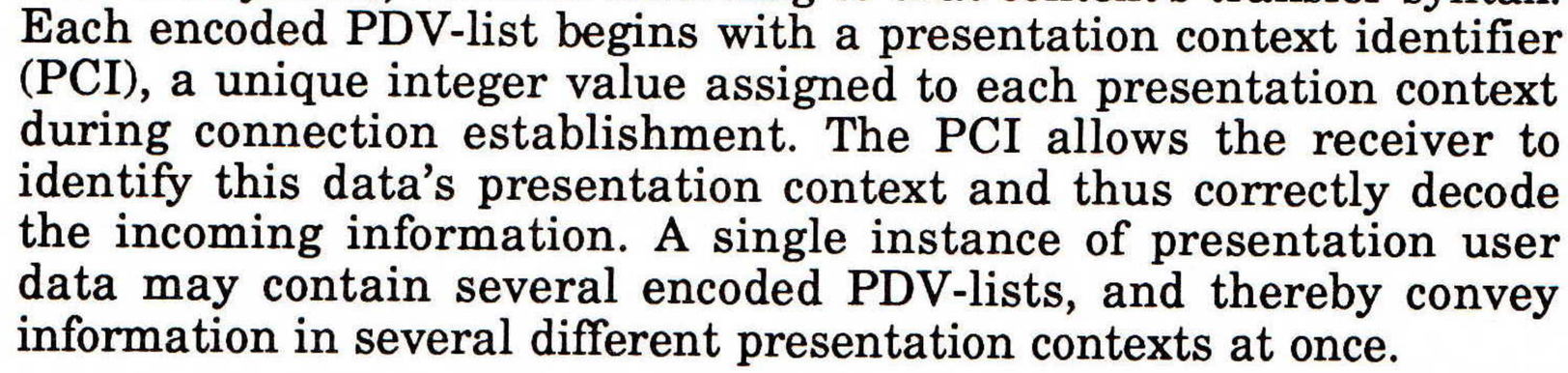


Figure 4: Presentation user data

Releasing connections

The presentation protocol provides three ways to release a connection: orderly release, user abort, and provider abort. If a presentation user wishes to ensure that no data is lost during connection release, that user will choose the orderly release option. In some cases, however, the presentation user may not care whether any data is lost, wishing instead just to end the communication. In this situation, that user may issue a user abort service, abruptly ending the presentation connection and possibly losing some of the data currently in transit. Finally, one or both of the presentation entities may detect some anomaly or error in communication. If this occurs, the provider of the presentation service (i.e., the presentation entities themselves) will indicate to their users that a provider abort has occurred and the presentation connection is gone.

Conclusion

The OSI presentation protocol itself is very straightforward, with none of the complex timers, flow control mechanisms, and retransmission schemes found in the lower layers. The concepts embodied in the layer's operation, however, are not so simple, and neither is the construction of software to implement this protocol and the associated encoding/decoding functions. And yet, by freeing application layer protocols from concern with the representation of transferred information, the presentation layer is nevertheless an important piece in the structure of OSI.

DAVID CHAPPELL has been active in the design and implementation of OSI protocols for the past several years as a software engineer with NCR Corporation and Cray Research. He has also been an active participant in the NIST OSI Workshops, and currently chairs the Workshop's Upper Layers Special Interest Group. David holds an M.S. in computer science from the University of Wisconsin, and now spends most of his time writing, consulting, and teaching about OSI and related topics.

Host Requirements RFCs have arrived!

The long awaited Host Requirements RFCs are now available from the Network Information Center in the online library at NIC.DDN.MIL.

RFC 1122: Requirements for Internet Hosts—Communication Layers. This RFC is an official specification for the Internet community. It incorporates by reference, amends, corrects, and supplements the primary protocol standards documents relating to hosts.

RFC 1123: Requirements for Internet Hosts—Application and Support. This RFC is an official specification for the Internet community. It incorporates by reference, amends, corrects, and supplements the primary protocol standards documents relating to hosts. Distribution of these documents is unlimited.

Getting RFCs

RFCs can be obtained by anonymous FTP from NIC.DDN.MIL, with the pathname RFC:RFCnnnn.TXT (where "nnnn" refers to the number of the RFC). The NIC also provides an automatic mail service for those sites which cannot use FTP. Address the request to SERVICE@NIC.DDN.MIL and in the subject field of the message indicate the RFC number, as in "Subject: RFC 1122."

CONNEXIONS
480 San Antonio Road
Suite 100
Mountain View, CA 94040
415-941-3399
FAX: 415-949-1779

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

CONNEXIONS

PUBLISHER Daniel C. Lynch

EDITOR Ole J. Jacobsen